

**APPLICATION PROGRAMMING INTERFACE
FOR MEASURING DEVICES**

BACKGROUND OF THE INVENTION

5 This application is being filed with a
microfiche appendix of computer program listings
consisting of one (1) fiche having fifty-one (51)
frames.

10 A portion of the disclosure of this patent
document contains material which is subject to
copyright protection. The copyright owner has no
objection to the facsimile reproduction by anyone of
the patent document or the patent disclosure, as it
15 appears in the Patent and Trademark Office patent
file or records, but otherwise reserves all
copyright rights whatsoever.

Field Of The Invention

20 The present invention concerns an
application programming interface (API) usable by a
software developer when writing an application
program, such as a color calibration program, that

uses a color measuring device. In particular, the present invention relates to an API that provides a common interface between such an application program and plural different types of color measuring devices.

Description Of The Related Art

One common use of color measuring devices is to calibrate color fidelity of color recording devices. Typically, the recording device (such as a color laser beam printer or ink jet printer) records color patches on a recording medium. The color values of the patches are measured by the color measuring device, the measured values are compared against expected (or ideal) values, and the comparisons are used to calibrate the recording device.

Color measuring devices come in many different physical structures, such as XY tables, strip readers, hand-held patch readers, and flatbed scanners, and operate in different ways using different numbers and types of sensors. Color patches are input in different ways to color measuring devices which have different physical structures. For example, patches are input to an XY table on a sheet that has an array of patches printed thereon, whereas patches are input to a strip reader on strips that each have one or more rows of patches printed thereon. The sensors that actually perform color measurements are positioned on patches in different ways depending on the physical structures of the color measuring devices.

In order to make accurate color measurements for use in calibrating a color recording device, the color measuring devices themselves must be calibrated properly. For example, for some color measuring devices to

position their sensor(s) accurately, the positioning mechanism in the color measuring devices must be calibrated. In addition, the sensors in some color measuring devices must be calibrated so as to ensure proper color measurements. Depending on the numbers and types of sensors as well as the physical structures of the color measuring devices, calibration of the position of the sensors and calibration of the sensors differ among different types of color measuring devices.

In addition, application programs that use color measuring devices should provide interfaces to the user. These interfaces should include instructions to the user at various points throughout operations performed using the color measuring devices. The timing and content of these instructions also differ among different color measuring devices.

As a result of the foregoing differences in operation among the different types of color measuring devices, a different command set is needed for each different type of device. Application programs that use the color measuring devices often must include different numbers of calls to commands in the different command sets in order to perform a same operation with different color measuring devices. In addition, the application programs often must include different interfaces for each different type of color measuring device. These different command sets and interfaces greatly complicate the task of writing application programs that perform color measurements. As a result, conventional application programs for performing color measurements must include separate and often lengthy routines for taking color measurements with each different type of supported device. In addition, once a conventional application program is

written, that application program cannot use new types of color measuring devices, because the new device would require calls to commands in a new command set.

5

SUMMARY OF THE INVENTION

Accordingly, there is a need for an application programming interface (API) that allows a software developer to write a single set of routines for taking color measurements using plural different types of color measuring devices.

The present invention addresses the foregoing need by providing an application programming interface that abstracts the operations needed to perform color measurements. An application program only needs to include calls to the functions in this API in order to use any supported type of color measuring device. The API in turn calls the necessary commands for performing color measurement operations.

In order to allow the application program to provide appropriate instructions during operations performed with a color measuring device, certain functions in the API are called numerous times in order to complete their respective operations. In order to enable the application program to call such a function a correct number of times, the API provides the application program with flow control data of the number of times that the function needs to be called so as to complete its operation with a particular color measuring device. The API also preferably provides the application program with display values for display so as to instruct the user in manipulating the color measuring device.

By virtue of the foregoing, the application program does not need to include separate routines

for each different type of supported color measuring device. Instead, the application program need only include calls to the functions in the API and include mechanisms for handling display values and flow control data provided by the API. Moreover, if the API is implemented in a dynamically linkable library (DLL), an existing application program that calls the functions in the API can use new types of color measuring devices by linking to an updated version of the DLL.

Accordingly, in one aspect the invention is an API that provides a common interface between an application program and plural different types of color measuring devices each having at least one color measuring sensor. The API includes plural functions for operating any of the plural different types of color measuring devices. In order to complete an operation performed by at least one of the plural functions, the function that performs the operation must be called a number of times which is different for at least two different types of color measuring devices. For a particular color measuring device, the API provides the application program with flow control data of the number of times that the function must be called. This flow control data preferably can be provided by the function, in the form of a call-again value or as a numerical value, or by a separate function in the API such as a get-device-capabilities function. In some embodiments of the invention, a combination of these methods of providing the flow control data is utilized.

Preferably, the functions in the API also provide the application program with display values which are different for at least two different types of color measuring devices. The display values are for display to a user, preferably by the application

In another aspect, the invention is an API that provides a common interface between an application program and plural different types of color measuring devices each having at least one color measuring sensor. The API includes plural functions for operating any of the plural different types of color measuring devices. The plural functions include a calibrate-position function, a calibrate-sensor function, a move-to-patch function, and a make-measurement function.

The calibrate-position function calibrates a relative position of a recording medium with respect to any of the plural different types of color measuring devices. Preferably, the calibrate-position function provides the application program with at least one display value that is to be displayed so as to instruct a user to position the recording medium or to position any of the color measuring sensors.

The calibrate-sensor function calibrates any of the color measuring sensors of any of the plural different types of color measuring devices. Preferably, the calibrate-sensor function provides the application program with at least one display value that is to be displayed so as to instruct the user in calibrating the sensor.

The move-to-patch function relatively
positions any of the color measuring sensors and a
color patch for any of the plural different types of
color measuring devices. The move-to-patch function
is provided with a logical color patch number by the
application program. In order to relatively
position the color patch and any of the color
measuring sensors for some color measuring devices,
the move-to-patch function preferably can provide

5
10

15

20

25

30

35

5

10

15

25

30

35

Figure 2 is a detailed block diagram showing the internal architecture of the computer system shown in Figure 1.

5 Figure 4 is a block diagram for explaining
a structure of a measuring device application
programming interface (API) according to the
invention.

Figure 6 is a detailed block diagram showing the internal architecture of the development system shown in Figure 5.

20 Figure 8 is a block diagram for explaining
a variation of the structure of the measuring device
API according to the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

35 Color printer 1 is depicted in Figure 1 as
an ink jet printer. However, any printer capable of
printing color images on recording media can be

utilized by a color calibration program written according to the invention, such as ink jet printers and laser printers. Color printer 1 prints color images in response to print jobs received from computer system 6 over printer interface 8.

Measuring device 10 is a color measuring device that is used to generate color measurements of color patches input thereto. The color measurements generated by color measuring device 10 are transmitted over measuring device interface 9 to computer system 6. Examples of measuring device 10 include XY table 2, strip reader 3, and hand-held patch reader 4 for reading individual color patches.

Color patches are input to each of the different types of color measuring devices in different ways. For example, patches are input to XY table 2 on a sheet that has an array of patches printed thereon, whereas patches are input to strip reader 3 on strips that each have one or more rows of patches printed thereon. The sensors that actually perform color measurements are positioned on patches in different ways depending on the physical structures of the color measuring devices. For example, XY table 2 typically includes motors and gimbals for automatically positioning the sensors, while hand-held patch reader 4 typically must be manually positioned over a patch.

Flatbed scanner 5 can also serve as a color measuring device, provided that scanner 5 is properly calibrated. Color patches are input to flatbed scanner 5 in a similar manner as the patches are input to XY table 2, namely on a sheet that has an array of patches printed thereon.

The different color measuring devices also can use entirely different types and arrangements of sensors for making the color measurements. For example, color measuring device 10 can be a

spectrometer (i.e., a spectrophotometer or a spectroradiometer), which uses sensors that preferably measure light intensity for frequencies across the entire visible spectrum. Alternatively, color measuring device 10 can be a colorimeter or densitometer, which uses sensors that measure light intensity for XYZ or RGB components of the spectrum.

In any case, each of the color measuring devices have at least one color sensor for making color measurements. In order to ensure accurate color measurements, the sensor(s) should be calibrated. In addition, depending on the type of measuring device 10, a relative position of a recording medium with respect to the device might have to be calibrated. In particular, each time a new recording medium (e.g., strip or sheet) is input to color measuring device 10, the position of the recording medium should be recalibrated so that the color measuring device can properly locate color patches printed thereon.

Because of the different ways that color patches are input and measured by the different types of color measuring devices, a different command set is needed to calibrate and to make color measurements with each different type of color measuring device.

As shown in Figure 1, computer system 6 is connected to printer 1 through printer interface 8 and measuring device 10 through measuring device interface 9. Provided with computer system 6 are display 12 which may be a color monitor, keyboard 14 for entering user commands, and pointing device 15 such as a mouse for pointing to and for manipulating graphical user interfaces and other objects displayed on display 12.

Computer system 6 also includes a mass storage device such as fixed disk 16. This mass

storage device is for storing computer-executable process steps for a color calibration program and a measuring device DLL according to the invention, as well other application programs and an operating system including a printer driver and a measuring device driver. Such storage may also be provided by a CD-ROM (not shown).

It should be understood that, although a programmable general-purpose computer is shown in Figure 1, a dedicated computer terminal or other type of data processing equipment can utilize the present invention.

Figure 2 is a detailed block diagram showing the internal architecture of computer system 6. As shown in Figure 2, computer system 6 includes central processing unit 23 which interfaces with computer bus 24. Also interfacing with computer bus 24 are fixed disk 16, main memory (RAM) 26, read only memory (ROM) 27, floppy disk interface 29, display interface 30 to display 12 (not shown), keyboard interface 32 to keyboard 14 (not shown), pointing device interface 33 to pointing device 15 (not shown), printer interface 8 to color printer 1, and measuring device interface 9 to measuring device 10.

Main memory 26 interfaces with computer bus 24 so as to provide RAM storage to CPU 23 during execution of software applications. More specifically, CPU 23 loads process steps from fixed disk 16, another storage device, or some other source such as a network (not shown), into main memory 26. CPU 23 then executes the stored process steps from main memory 26 in order to execute application programs. Data such as print data and color measurement data can be stored in main memory 26, where the data can be accessed by CPU 23 during execution of the process steps.

As shown in Figure 2, fixed disk 16 typically contains operating system 36, application programs 37, measuring device DLL 38 according to the invention, and other files and data. Operating system 36 includes printer driver 40 and measuring device driver 41. Measuring device driver 41 is one of plural different types of measuring device drivers corresponding to the type of measuring device 10 connected to computer system 6. Examples of measuring device driver 41 include XY table driver 43 for XY table 2, strip reader driver 44 for strip reader 3, patch reader driver 45 for hand-held patch reader 4, and scanner driver 46 for scanner 5. Some or all of these drivers can be stored on fixed disk 16.

Application programs 37 include color calibration program 48. The structure and operation of color calibration program 48 is explained in more detail below with reference to Figure 3.

Measuring device DLL 38 is a library of functions that are called by an application program so as to use any of plural different color measuring devices supported by the DLL. In order to write an application program that calls these functions, a software developer uses a measuring device API according to the invention, as explained below with reference to Figures 5 and 6.

Figure 3 is a block diagram for explaining a system for calibration of color fidelity of a color printer by a color calibration program according to the invention. In Figure 3, color calibration program 48 runs on computer system 6 and sends print data 51 for printing color patches to printer driver 40. The color patches are for use in calibration of color printer 1.

Color calibration program 48 makes API calls 52 to functions provided by measuring device

DLL 38 in order to make color measurements of the color patches. According to the invention, since the DLL encapsulates the functionality needed to operate many different color measuring device, it is possible for the color calibration program 48 to call exactly the same functions, in exactly the same order, no matter which of plural different types of color measuring devices are connected. Because of the difference in devices, however, the API often needs to have certain functions repeated, or operator interaction performed, in order for the different devices to be usable by a single application. Accordingly, in response to API calls 52, the functions in the DLL provide color calibration program 48 with information that the application needs to ensure that the API functions are called correctly. Such information might include, for example, display values for display by the application to the user so as to prompt the user for some user intervention (such as to prompt the user to insert a next sheet of patches into the measuring device), and the information might further include flow control data that instruct the application to alter its control flow such as by calling particular DLL functions multiple times until a needed result (such as position calibration) is achieved. Display values and flow control data 53 provided to color calibration program 48 are explained in more detail below with reference to Figures 4, 8 and 9.

Color calibration program 48 also is provided with color measurements 54 by the DLL. Based on color measurements 54, color calibration program 48 sends color adjustment commands 55 to printer driver 40 so as to adjust color fidelity of color printer 1. The operation of color calibration program 48 in printing and measuring the color

patches and adjusting the color printer is explained in more detail below with reference to Figures 7A to 7C.

In response to print data 51 received from color calibration program 48, printer driver 40 sends print job 57 to color printer 1. Likewise, in response to color adjustment commands 55 received from color calibration program 48, printer driver 40 sends printer commands 58 to color printer 1.

Color printer 1 prints printed color patches 61 in response to print job 57 received from printer driver 40. The user inputs these color patches to color measuring device 10.

Measuring device DLL 38 provides a common interface between color calibration program 48 and plural different types of color measuring devices. As mentioned above, functions in measuring device DLL 38 are called by API calls 52 from color calibration program 48. The functions in measuring device DLL 38 make device driver calls 62 based on API calls 52 in order to operate measuring device 10. Device driver calls 62 are calls to a device driver corresponding to the type of color measuring device 10. For example, if measuring device 10 is an XY table, device driver calls 62 are calls to an XY table driver. In response to device driver calls 62, measuring device DLL 38 is provided with color measurements 64 by measuring device driver 41. Measuring device DLL 38 in turn provides color calibration program 48 with display values and flow control data 53 and color measurements 54. The flow control data preferably is based at least in part on device characteristics 65 received from measuring device driver 41. The structures of the functions in measuring device DLL 38 according to the invention are explained in more detail below with reference to Figures 4, 8 and 9, and the

operation of the functions is explained with reference to Figures 7A to 7C.

Measuring device driver 41 is a device driver corresponding to the type of measuring device 10. Measuring device driver 41 receives device driver calls 62 from the functions in measuring device DLL 38. In response to these call, measuring device driver 41 sends device commands 66 to measuring device 10, which provides color measurements 67 to measuring device driver 41. Color measurements 67 in turn are provided to measuring device DLL 38 as color measurements 64.

As mentioned above, measuring device 10 receives printed color patches 61 as input. Measuring device 10 makes color measurements 67 of printed color patches 61 in accordance with device commands 66 received from measuring device driver 41. Measurement device 10 provides color measurements 67 to measuring device driver 41.

Figure 4 is a block diagram for explaining functionality provided in the DLL and accessed from the application through the API.

Briefly, the measuring device DLL illustrated in Figure 4 is accessed through a measuring device API according to the invention. The API provides a common interface between an application program and plural different types of color measuring devices each having at least one color measuring sensor. The interface includes plural functions for operating any of the plural different types of color measuring devices. In order to complete an operation performed by one or more of the functions, the function that performs the operation must be called a number of times which is different for different types of color measuring devices. For a color measuring device that is being operated, the API provides the application program

5 This flow control data is provided for other
functions by the functions themselves, in the form
of "call-again" values provided by the functions to
the color calibration program. The functions in the
API also preferably provide the application program
10 with display values which are different for
different types of color measuring devices. The
values are displayed by the color calibration
program to a user so as to instruct the user in
manipulating the color measuring device that is
15 being operated.

In more detail, Figure 4 shows measuring device DLL 38 which is accessed by an application program through a measuring device API according to the invention. Thus, the API provides an interface for a software developer to write an application program that calls the functions in the DLL illustrated in Figure 4. The functions provide an interface between color calibration program 48 and measuring device driver 41. Measuring device driver 41 in turn communicates with measuring device 10 through measuring device interface 9.

Measuring device driver 41 is a device driver corresponding to the type of measuring device 10. Thus, brackets next to measuring device driver 41 in Figure 4 indicate that measuring device driver 41 is one of plural different device drivers such as XY table driver 43, strip reader driver 44, patch reader driver 45, and scanner driver 46. Likewise, brackets next to measuring device 10 in Figure 4 indicate that measuring device 10 is one of XY table 2, strip reader 3, hand-held patch reader 4 for reading individual color patches, and scanner 5.

It should be noted that the various drivers and devices shown in Figure 4 are only representative of the possible drivers and devices that can be operated using the API according to the invention.

In Figure 4, measuring device DLL 38 includes at least the following functions, which are provided to an application program through a measuring device API according to the invention: get-device-capabilities 71, calibrate-position 72, calibrate-sensor 73, move-to-patch 74, and make measurement 75. Arrows pointing into blocks for these functions indicate information provided to the functions by color calibration program 48. Likewise, arrows pointing out of blocks for these functions indicate information provided by the functions to color calibration program 48.

Information can be provided in various different ways to the functions by color calibration program 48 and by the functions to color calibration program 48. For example, information can be provided to a function as a passed variable, by modification of a shared (e.g., global) variable or data space, by passing a pointer to a memory location that stores the information, or through messages in a message-based environment. Likewise, information can be provided by a function as a returned value, by modification of a shared variable or data space, by modification of a memory location pointed to by a pointer, or through messages. Any or all of these methods, as well as any other additional methods, can be utilized by measuring device DLL 38 according to the invention.

Get-device-capabilities function 71 according to the invention provides color calibration program 48 with flow control data for calibrate-position function 72 and calibrate-sensor

function 73. In particular, get-device-capabilities
function 71 provides color calibration program 48
with number of calls 77, thereby indicating the
number of calls that must be made to calibrate-
5 position function 72 in order to calibrate a
position of a recording medium with respect to
measuring device 10. Likewise, get-device-
capabilities function 71 provides color calibration
program 48 with number of calls 78, thereby
10 indicating the number of calls that must be made to
calibrate-sensor function 73 in order to calibrate
the sensor(s) of measuring device 10.

The DLL preferably determines the flow
control data by querying measuring device driver 41,
15 or measuring device 10, for device characteristics
65. The functions in measuring device DLL determine
the flow control data based at least in part on
device characteristics 65. For example, if the
device characteristics identify the type of
20 measuring device 10, the functions can determine the
flow control data using a look-up table.
Alternatively, device characteristics 65 can include
the flow control data directly, for later use by the
different functions in the DLL. Get-device-
25 capabilities function 71 preferably queries the
device driver or the device for the device
characteristics.

Calibrate-position function 72 is called to
calibrate a relative position of a recording medium
30 with respect to any of the plural different types of
color measuring devices. In order to calibrate the
relative position of the recording medium,
calibrate-position function 72 might have to be
called more than once. The number of calls depends
35 at least in part on the type of measuring device 10.
As discussed above, get-device-capabilities function
71 according to the invention provides color

calibration program 48 with number of calls 77 that must be made to calibrate-position function 72 in order to complete the position calibration operation. Number of calls 77 preferably is based at least in part on device characteristics 65.

Each time calibrate-position function 72 is called, the user might be required to perform a task such as positioning the sensor(s), positioning the recording medium, or otherwise manipulating measuring device 10. Calibrate-position function 72 preferably provides display values 80 to color calibration program 48, and program 48 displays display values 80 to a user so as to instruct the user to perform these tasks, as necessary.

Display values 80, as well as all other display values provided by measuring device DLL 38, can include text and/or graphics, or any other type of values displayable by computer system 6. Because the application program actually displays the display values, the application program can determine the form of the display. For example, the application program can display text or graphics based on the display values. The display values do not necessarily need to be displayed visually. For example, if the application program includes text-to-speech functionality, the program can generate speech based on the display values.

Calibrate-sensor function 73 is called to calibrate the sensor(s) of any of the plural different types of color measuring devices. In order to calibrate the sensor(s), calibrate-sensor function 73 might have to be called more than once. As discussed above, get-device-capabilities function 71 according to the invention provides color calibration program 48 with number of calls 78 that must be made to calibrate-sensor function 73 so as to complete the sensor calibration operation.

Each time calibrate-sensor function 73 is called, the user might be required to perform a task such as manipulating measuring device 10. Calibrate-sensor function 73 preferably provides display values 82 to color calibration program 48 for display by the program to a user so as to instruct the user to perform any such tasks.

Move-to-patch function 74 is called to relatively position the sensor(s) and a color patch for any of the plural different types of color measuring devices. Move-to-patch function 74 is provided with logical color patch number 84 by color calibration program 48. In response, move-to-patch function 74 moves the sensor(s) or the recording medium so as to position the sensor(s) at the corresponding color patch, provides color calibration program 48 with display values 85 so as to instruct the user to move the sensor(s) or the recording medium, or both.

In order to relatively position the sensor(s) at the patch, move-to-patch function 74 might have to be called multiple times, depending on the type of measuring device 10. For example, if the color patch corresponding to color patch number 84 is on a next sheet or strip of recording medium, move-to-patch function 74 might have to be called a first time so as to instruct a user to change the sheet or strip, a second time so as to recalibrate the position of the recording medium (by providing a recalibrate value), and then a third time so as to move to the first patch on the recording medium. The number of times that move-to-patch function 74 needs to be called preferably is determined by the function based at least in part on device characteristics 65. If move-to-patch

5

10

15

25

35

takes the form of call-again value 90, which is provided if make-measurement function 75 has not been called the necessary number of times.

If the sensor(s) need to be recalibrated after a color measurement, for example after a certain number of color measurements, make-measurement function 75 provides color calibration program 48 with recalibrate value 91.

Figure 5 is a representational view of a development system for development of an application program using a measuring device API according to the invention. As shown in Figure 5, development system 100 is provided with display 102 which may be a color monitor, keyboard 104 for entering user commands and for entering code for an application program, and pointing device 105 such as a mouse. Development system 100 also includes a mass storage device such as fixed disk 106. This mass storage device is for storing computer-executable process steps for an operating system, software development programs, and a measuring device API according to the invention. Such storage may also be provided by a CD-ROM (not shown).

It should be understood that, although development system 100 is shown as a programmable general-purpose computer in Figure 5, a dedicated computer terminal or other type of data processing equipment can be used to write application programs utilizing the API according to the present invention.

Figure 6 is a detailed block diagram showing the internal architecture of development system 100. As shown in Figure 6, development system 100 includes central processing unit 113 which interfaces with computer bus 114. Also interfacing with computer bus 114 are fixed disk 106, main memory (RAM) 116, read only memory

(ROM) 117, floppy disk interface 119, display interface 120 to display 102 (not shown), keyboard interface 122 to keyboard 104 (not shown), and pointing device interface 123 to pointing device 105 (not shown).

Main memory 116 interfaces with computer bus 114 so as to provide RAM storage to CPU 113 during execution of software applications. More specifically, CPU 113 loads process steps from fixed disk 106, another storage device, or some other source such as a network (not shown), into main memory 116. CPU 113 then executes the stored process steps from main memory 116 in order to execute application programs. Data such as code for an application program utilizing a measuring device API according to the invention can be stored in main memory 116, where the data can be accessed by CPU 113 during execution of the process steps.

As shown in Figure 6, fixed disk 106 typically contains operating system 126, application programs 127, and measuring device API 128 according to the invention. Operating system 126 typically is a windowing operating system and preferably is compatible with operating system 36 of computer system 6. Application programs 127 preferably includes software development programs 130 for developing application programs for making color measurements with color measuring devices.

A software developer uses software development programs 130 to write code for an application program such as color calibration program 48. The software developer includes calls to functions provided by measuring device API 128 in order to have the application program make color measurements with any color measuring device supported by the API.

The API preferably includes headers corresponding to the functions in measuring device DLL 38. When the software developer compiles the code for application program using software development programs 130, these headers allow the application program to access the functions in measuring device DLL 38. These functions provide a common interface to any of the plural supported color measuring devices. Therefore, as long as the application program includes code for handling flow control data and display values returned by the functions in the DLL, as described above with reference to Figures 3 and 4, the application program is able to make color measurements with any of the plural supported color measuring devices.

Figures 7A to 7C are flowcharts for explaining a representative color calibration application using a measuring device API according to the invention.

In step S701 in Figure 7A, color calibration program 48 generates print data 51 for color patches for calibration of color fidelity of color printer 1. In step S702, color calibration program 48 sends print data 51 to printer driver 40, which in turn sends print job 57 to color printer 1 based on print data 51. Color printer 1 prints the print job onto a recording medium, resulting in printed color patches 61.

In step S703, color calibration program 48 makes API call 52 to get-device-capabilities function 71 in measuring device DLL 38. In response, get-device-capabilities function 71 provides flow control data to calibration program 48. This flow control data preferably includes number of calls 77 and number of calls 78, which indicate the number of times calibrate-position function 71 and calibrate-sensor function

72 need to be called in order to perform their respective operations.

5 In step S704, the relative position of the sensor(s) with respect to the recording medium is calibrated. Figure 7B is a flowchart for explaining position calibration according to the invention.

10 In step S705 of Figure 7B, color calibration program 48 calls calibrate-position function 72. This function sends device driver calls 62 to measuring device driver 41, which in turn sends device commands 66 to measuring device 10 so as to calibrate the relative position of the sensor(s).

15 In step S706, color calibration program 48 determines if calibrate-position function 72 provided display values 80. If calibrate-position function 72 did provide display values 80, color calibration program 48 displays the display values to the user, such as on display 12, in step S707.

20 In step S708, color calibration program 48 determines if calibrate-position function 72 has been called the number of times indicated by number of calls 77. If calibrate-position function 72 has not been called number of calls 77
25 times, flow returns to step S705. Once calibrate-position function 72 has been called number of calls 77 times, flow returns to Figure 7A.

30 In Figure 7A, flow proceeds to step S709, where the sensor(s) are calibrated. Figure 7C is a flowchart for explaining sensor calibration according to the invention.

35 In step S710 of Figure 7C, color calibration program 48 calls calibrate-sensor function 72. This function sends device driver calls 62 to measuring device driver 41, which in turn sends device commands 66 to measuring device 10 so as to calibrate the sensor(s).

66000-25871060

5

10

15

25

35

5

10

15

20

25

30

35

5

10

15

20

30

35

Because each of the functions in measuring device DLL 38 accessed through measuring device API 128 makes device driver calls 62 to measuring device 41 for the type of color measuring device 10, color calibration program 48 is insulated from the differences between different types of color measuring devices. Furthermore, the flow control data for multiple calls provided to color calibration program 48 allows the program to call the functions the necessary number of times for each operation without knowing the type of the actual device. The display values provided to the program allow the program to display the display values appropriate for the type of measuring device 10 that is being used. Moreover, since the application program generates displays based on the display values rather than the DLL, the application program can package a message however it wants so as to permit each developer to give his application a unique and internally-consistent GUI. As a result, a block of code that accesses color measuring device DLL 38 can make color measurements with any type of color measuring device supported by the API, despite the different device driver commands needed to calibrate and operate the different types of devices.

Included in a microfiche appendix submitted herewith is sample code that operates generally as described above. This code includes code for a portion of a color calibration program that makes color measurements using an API according to the invention, the API itself, and a DLL accessed through the API. This code is provided for illustrative purposes only. Other implementations of the color measurement code, measuring device API, and measuring device DLL exist that are within the scope and spirit of the invention.

Figure 8 is a block diagram for explaining a variation of the structure of the measuring device DLL according to the invention and of the API used to access the DLL. As shown in Figure 8, both
5 calibrate-position function 72 and calibrate-sensor function 73 can provide call-again values 140 and 141, respectively, to color calibration program 48. These call-again values are provided by
10 these functions when they need to be called more than once in order to complete their respective operations.

Because the calibration functions can return these call-again values to facilitate multiple calls, there is no need for get-device-
15 capabilities function 71 to provide number of calls 77 and number of calls 78 to the calibration functions. Of course, get-device-capabilities function 71 can still be included in the API, if desired.

Figure 9 is a block diagram for explaining another variation of the structure of the measuring device API according to the invention. In this variation, each of the functions in the API provides
20 flow control data in the form of a "number of calls" value to indicate how many times that function should be called in order to complete its respective operation. Thus, calibrate-position function 72 provides number of calls 143, calibrate-sensor
25 function 73 provides number of calls 154, move-to-patch function 74 provides number of calls 155, and make-measurement function 75 provides number of calls 156. These values can be a total number of calls that need to be made to each function. For
30 example, if three calls need to be made to calibrate-position function 72, then that function would provide a value of "three" each time it was called. Alternatively, each function can maintain
35

662020-2534200

5
10

15

20